"Express Mail" mailing label number: EV 164034248 US

Date of Deposit: January 27, 2004

Attorney Docket No.15396US01

SYSTEM, AND METHOD FOR CALCULATING PRODUCT OF CONSTANT AND MIXED NUMBER POWER OF TWO

RELATED APPLICATIONS

[001] [Not Applicable]

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[002] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[003] [Not Applicable]

BACKGROUND OF THE INVENTION

[004] An MPEG audio decoder decodes compressed audio data by performing a number of different functions and computations. Among the functions and computations is rescaling of frequency coefficients. The rescaling of frequency coefficients comprises the multiplication of constants, C, by a power of 2, 2^X, where X comprises the sum of an integer and a fraction.

[005] Computation of 2^f in hardware is complex where f is a non-integer. Although the computation of 2^f is less complex in software, speed considerations involved in decompressing audio data in real time makes a software solution less desirable.

[006] Further limitations and disadvantages of conventional and traditional approaches will become

apparent to one of skill in the art through comparison of such systems with embodiments presented in the remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE INVENTION

[007] Presented herein are systems and methods for computing the product of a constant and a mixed number power of two.

[008] In one embodiment, there is a circuit for computing a product of a constant and a mixed number power of two. The mixed number comprises an integer portion and a fraction portion. The circuit comprises a first register, a second register, a memory, a third register, and a multiplier circuit. The first register stores the constant. The second register stores the integer portion and the fraction portion. The memory stores a plurality of values, each of said plurality of values corresponding to a particular one of a corresponding plurality of fractions, wherein each one of said plurality of values is two to the exponential fraction corresponding to the one of said plurality of values. The third register stores a particular one of the plurality of values, said particular one of the plurality of values corresponding to the fraction portion. The multiplier circuit multiplies the contents of the third register by the contents of the first register, thereby resulting in a product. The product is shifted a certain number of times, the certain number of times equal to the integer portion.

[009] In another embodiment, there is presented a method for computing a product of a constant and mixed number power of two. The mixed number comprises an integer portion and a fraction portion. The method comprises receiving the constant; receiving the integer portion; receiving the fraction portion; providing the fraction portion to a memory storing a plurality of values, each of said plurality of values corresponding to a particular one

of a corresponding plurality of fractions, wherein each one of said plurality of values is two to the exponential fraction corresponding to the one of said plurality of values; receiving a particular one of the plurality of values, said particular one of the plurality of values corresponding to the fraction portion; multiplying the constant by the particular one of the plurality of values, thereby resulting in a product; and shifting the product a certain number of times, the certain number of times equal to the integer portion.

[0010] In another embodiment, there is presented a audio decoder for decoding compressed audio data. The audio decoder comprises a Huffman decoder, inverse quantizer and a resampler. The Huffman decoder decodes quantized coefficients. The inverse quantizer inverse quantizes the quantized coefficients. The rescaler multiplies the un-scaled inversely quantized coefficients with relevant scale factors. The rescaler comprises a first register, a second register, a memory, a third register, and a multiplier circuit. The first register stores the constant. The second register stores the integer portion and the fraction portion. The memory stores a plurality of values, each of said plurality of values corresponding to a particular one of a corresponding plurality of fractions, wherein each one of said plurality of values is two to the exponential fraction corresponding to the one of said plurality of values. The third register stores a particular one of the plurality of values, said particular one of the plurality of values corresponding to the fraction portion. The multiplier circuit multiplies the contents of the third register by the contents of the first register, thereby resulting in a product. The product is shifted a certain

number of times, the certain number of times equal to the integer portion.

[0011] These and other advantages and novel features of the present invention, as well as details of an illustrated embodiment thereof, will be more fully understood from the following description and drawings.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0012] **FIGURE 1** is a block diagram describing an exemplary circuit for calculating the product of a coefficient and a mixed number power of two, in accordance with an embodiment of the present invention;

[0013] **FIGURE 2** is a flow diagram for calculating the product of a coefficient and a mixed number power of two, in accordance with an embodiment of the present invention;

[0014] **FIGURE 3** is a block diagram describing the compression of an audio signal;

[0015] **FIGURE 4** is a block diagram of an exemplary decoder system in accordance with an embodiment of the present invention; and

[0016] **FIGURE 5** is a block diagram of an audio decoder in accordance with an embodiment of the present invention; This figure is not relevant.

DETAILED DESCRIPTION OF THE INVENTION

[0017] Referring now to **FIGURE 1**, there is illustrated a block diagram of a circuit for calculating the product of a coefficient, C, and a mixed number X power of two. The coefficient can comprise any number, both an integer, or mixed number, and can be positive or negative. The mixed number X is the sum of an integer portion N, and a fraction portion f.

[0018] The coefficient C is received in a coefficient register 105 and the integer portion N and fraction portion f are received in an exponent register 110. The exponent register 110 comprises a first set of bits 110a for the integer portion N, and a second set of bits 110b for the fraction portion f.

[0019] The value $2^X C$ is equal to $2^N(2^f C)$. Accordingly, the product can be calculated by first multiplying the coefficient by the fraction power of two, and then multiplying the product by the integer power of two. Computation of 2^f in hardware is complex where f is a non-integer. Although the computation of 2^f is less complex in software, speed considerations may make a software solution less desirable. Instead, the value of 2^f, for a number of values of f, can be pre-calculated and stored in a memory 115.

[0020] As noted above, the second set of bits 110b store the fraction portion f. The memory 115 can store the value 2^f for each f that can be represented by the second set if bits 110b. For example, where the second set of bits comprises two bits, the two bits can represent, in decimal, f = 0, f = ¼, f = ½, and f = ¾. Accordingly, the memory 115

can store the values for 2^0 (1.0000), $2^{1/4}$ (1.1892), $2^{1/2}$,(1.4142) and $2^{3/4}$ (1.6818).

[0021] The memory 115 receives the fraction portion f from the second set of bits 110b of the exponent register 110. Upon receiving the second set of bits 110b, the memory 115 outputs the value for 2^f into another register 120.

[0022] The register 120 provides the value for 2^f to a multiplier circuit 125. The multiplier circuit 125 also receives the constant C from the constant register 105. The multiplier circuit 125 comprises a hardware circuit for multiplying operands, and can include, but is not limited to, an arithmetic logic unit. The multiplier circuit 125 multiplies the constant C from the constant register 105 and the value of 2^f from register 120, and writes the product 2^fC into a shift register 130.

[0023] As noted above, 2^XC is equal to $2^N(2^fC)$. The product of 2^N and 2^fC is computer from 2^fC by shifting in zeroes from the right side, N times. As the shift register 130 shifts in zeros from the right, the first set of bits 110a are decremented after each shift. When the first set of bits 110a equal zero, the shift register 130 stores 2^XC.

[0024] Referring now to **FIGURE 2**, there is illustrated a flow diagram for computing 2^XC, where X equals a mixed number that is the sum of an integer N, and a fraction f. The coefficient C is received at 205, the integer portion N is received at 210, and fraction portion f is received at 215.

[0025] The fraction portion f is provided to memory 115 at 218. Upon receiving the fraction portion f, the memory 115 provides (220) the value for 2^f. At 225, the constant C from the constant register 105 and the value of 2^f are multiplied, resulting in a product 2^fC.

[0026] The product of 2^N and 2^fC is computed from 2^fC by shifting in zeroes to the right side, N times. At 230, a zero is shifted in from the right side to the product 2^fC. At 235, the integer portion N is decremented. At 240, a determination is made whether N = 0. If at 240, N does not equal 0, 230 and 235 are repeated. If at 240, N = 0, then the product 2^fC with the zeroes shifted in from the right side equals 2^XC.

[0027] The present invention can be used for the calculation of a variety of non-linear functions in a variety of applications. For example, in MPEG-2 AAC decoders and MPEG-1 Layer-3 Decoders, spectral values are Huffman coded. In the decoder, the spectral values are Huffman decoded and need to be multiplied by 2^X (where X is a mixed number comprising the sum of an integer N, and a fraction f) in the inverse quantization block.

[0028] Referring now to **FIGURE 3**, there is illustrated a block diagram describing the encoding of an exemplary audio signal A(t). Pursuant to the MPEG-2 Advanced Audio Coding (MPEG-2 AAC) standard, the audio signal is sampled at rates starting at 8K samples/sec to 96K samples/sec. The samples are grouped into frames $F_0 \ldots F_n$ of 1024 samples, e.g., $F_x(0) \ldots F_x(1023)$. The frames $F_0 \ldots F_n$ are grouped into windows $W_0 \ldots W_n$ that comprise 2048 samples, e.g., $W_x(0) \ldots W_x(2047)$. However, each window W_x has a 50% overlap with the previous window W_{x-1}. Accordingly, the first 1024 samples of a window W_x are the same as the last 1024 samples of the previous window W_{x-1}. A window function w(t) is applied to each window $W_0 \ldots W_n$, resulting in sets $wW_0 \ldots wW_n$ of 2048 windowed samples, e.g., $wW_x(0) \ldots wW_x(2047)$. The modified discrete cosine transformation (MDCT) is applied to each set $wW_0 \ldots wW_n$ of windowed samples $wW_x(0) \ldots wW_x(2047)$, resulting

sets $MDCT_0 \ldots MDCT_n$ of 1024 frequency coefficients, e.g., $MDCT_x(0) \ldots MDCT_x(1023)$.

[0029] The sets of frequency coefficients $MDCT_0 \ldots MDCT_n$ are then quantized and coded for transmission, forming what is known as an audio elementary stream AES. The AES is then placed in fixed size transport packets, forming what is known as the audio transport stream (TS). The audio TS can be multiplexed with other audio TS and video TS. The multiplexed signal can then be stored, and/or transported for playback on a playback device. The playback device can either be local or remotely located. Where the playback device is remotely located, the multiplexed signal is transported over a communication medium, such as the internet. During playback, the Audio TSs are demultiplexed, resulting in the constituent AES signals. The constituent AES signals are then decoded, resulting in the audio signal.Referring now to **FIGURE 4**, there is illustrated a block diagram of an exemplary decoder for decoding compressed video data, configured in accordance with an embodiment of the present invention. A processor, that may include a CPU 490, reads a stream of transport packets 365b (a transport stream) into a transport stream buffer 432 within an SDRAM 430. The data is output from the transport stream presentation buffer 432 and is then passed to a data transport processor 435. The data transport processor then demultiplexes the MPEG transport stream into its PES constituents and passes the audio transport stream to an audio decoder 460 and the video transport stream to a video transport processor 440. The audio data is sent to the output blocks and the video is sent to a video decoder 445 and display engine 450.

[0030] Referring now to **FIGURE 5**, there is illustrated a block diagram describing an exemplary audio decoder 460 in accordance with an embodiment of the present invention. Once the frame synchronization is found, the AAC bitstream is demultiplexed by a bitstream demultiplexer 305. The bitstream demultiplexer separates the parts of the MPEG-2 AAC data stream into the parts for each tool, and provides each of the tools with the bitstream information related to that tool. The AAC decoder includes Huffman decoding 310, a rescaler 315, and the decoding of the side information used in tools such as mono/stereo 320, intensity stereo 325, TNS 330, and the filter bank 335. The sets of frequency coefficients $MDCT_0...MDCT_n$ are decoded and copied to the output buffer in a sample fashion. After Huffman decoding 310, an inverse quantizer 340 inverse quantizes each set of frequency coefficients $MDCT_0...MDCT_n$ by a 4/3 power nonlinearity. The rescaler 315 multiplies un-scaled inversely quantized frequency coefficients $MDCT_0...MDCT_n$ with scale factors.

[0031] Additionally, tools including the mono/stereo 320, intensity stereo 325, TNS 330, and can apply further functions to the sets of frequency coefficients $MDCT_0...MDCT_n$. Finally, the filter bank 335 transforms the frequency coefficients $MDCT_0...MDCT_n$ into the time domain signal A(t). The filter bank 335 transforms the frequency coefficients by application of the Inverse MDCT (IMDCT), the inverse window function, window overlap, and window adding.

[0032] The rescaler 315 multiplies the non-zero values by 2^X, where X comprises the sum of an integer N, and a fraction f. The rescaler 315 can comprise the circuit of **FIGURE 1** for multiplying the non-zero values by 2^X.

Alternatively, the rescaler 315 can multiply the non-zero values by 2^x by effectuating the flow diagram described in **FIGURE 2**.

[0033] The decoder system as described herein may be implemented as a board level product, as a single chip, application specific integrated circuit (ASIC), or with varying levels of the decoder system integrated with other portions of the system as separate components. The degree of integration of the decoder system will primarily be determined by the speed and cost considerations. Because of the sophisticated nature of modern processor, it is possible to utilize a commercially available processor, which may be implemented external to an ASIC implementation. Alternatively, if the processor is available as an ASIC core or logic block, then the commercially available processor can be implemented as part of an ASIC device wherein various operations are implemented in firmware.

[0034] While the invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the invention. In addition, many modifications may be made to adapt particular situation or material to the teachings of the invention without departing from its scope. Therefore, it is intended that the invention not be limited to the particular embodiment(s) disclosed, but that the invention will include all embodiments falling within the scope of the appended claims.